R_Analytics_Amazon

Calambro_Leysa

2024-11-10

Extracting Amazon Product Reviews

```
#4. Select 5 categories from Amazon and select 30 products from each category.



install.packages("rvest")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(rvest)
library(polite)
```

```
#phones category
url <- "https://www.amazon.com/s?k=Phone&crid=1NDRK9GG5A6SC&sprefix=phone%2Caps%2C988&ref=nb_sb_noss_1"

session <- bow(url,
user_agent = "Student's Demo Educational")
session
```

```
## <polite session> https://www.amazon.com/s?k=Phone&crid=1NDRK9GG5A6SC&sprefix=phone%2Caps%2C988&ref=n
##     User-agent: Student's Demo Educational
##     robots.txt: 138 rules are defined for 5 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
session_page <- scrape(session)
```

```
div_elements <- html_nodes(session_page, 'sg-col-20-of-24 s-matching-dir sg-col-16-of-20 sg-col sg-col-

# Create empty vectors to store data
links <- character()
img_srcs <- character()
titles <- character()
prices <- character()
ratings <- character()
descriptions <- character()
reviews <- character()
```

```
#5. Extract the price, description, ratings and reviews of each product.



for (div_element in div_elements) {
# Find the a element with class="a-link-normal s-no-outline" and get the link
```

```r
a_element <- html_node(div_element, 'a.a-link-normal s-line-clamp-2 s-link-style a-text-normal')
link <- ifelse(!is.na(a_element), paste0("https://www.amazon.com", html_attr(a_element, "href")), '')


# Find the img element with class="s-image" and get the link
img_element <- html_node(div_element, 'img.s-image')
img_src <- ifelse(!is.na(img_element), html_attr(img_element, "src"), '')

# Find the span element with class="a-size-base-plus a-color-base a-text-normal" and get the title
title_element <- html_node(div_element, 'h2.a-size-medium a-spacing-none a-color-base a-text-normal')
title <- ifelse(!is.na(title_element), html_text(title_element), '')

# Find the span element with class="a-price-whole" and get the price
price_element <- html_node(div_element, 'span.a-price-whole')
price <- ifelse(!is.na(price_element), html_text(price_element), '')

# Find the span element with class="a-icon-alt" and get the ratings
rating_element <- html_node(div_element, 'span.a-icon-alt')
rating <- ifelse(!is.na(rating_element), html_text(rating_element), '')
rating <- gsub("out of 5 stars", "", rating, fixed=TRUE)

description_element <- html_node(div_element, 'h2.a-size-base-plus a-text-bold')
description <- ifelse(!is.na(description_element), html_text(description_element), '')

review_element <- html_node(div_element, 'div.a-expander-collapsed-height a-row a-expander-container a-
review <- ifelse(!is.na(review_element), html_text(review_element), '')

# Append data to vectors
links <- c(links, link)
img_srcs <- c(img_srcs, img_src)
titles <- c(titles, title)
prices <- c(prices, price)
ratings <- c(ratings, rating)
descriptions <- c(descriptions, description)
reviews <- c(reviews, review)
}
```

```r
# Create a data frame
Phone_Products <- data.frame(Links = links,
Images = img_srcs,
Title = titles,
Price = prices,
Rating = ratings,
Description = descriptions,
Review = reviews)
Phone_Products
```

```
## [1] Links       Images      Title       Price       Rating      Description
## [7] Review
## <0 rows> (or 0-length row.names)
```

```r
write.csv(Phone_Products, "Phone_Products.csv")
```